FRONTSCHOOL

GIT CHEAT SHEET

Podstawowe komendy

Pierwsze kroki

> git config --global

Podstawowa konfiguracja git.
`git config --global (user.name | user.email)`

> git init

Utworzenie repozytorium wewnątrz folderu.

> git --version

Sprawdzenie zainstalowanej wersji git. Błąd oznacza, że git nie jest poprawnie zainstalowany.

> git help (-a) (*komenda*)

Uruchomienie pomocy dotyczącej komend.

Zapisywanie zmian

> git status

Aktualny stan repozytorium. Używamy głównie w celu sprawdzenia które pliki zostaną umieszczone w commicie.

> git commit (-a) -m *tytuł*

Utworzenie nowego commita w repozytorium.

> git log (--oneline)

Wyświetlenie historii commitów na aktualnym branchu.

> git add *file*|.

Dodanie pliku lub plików do poczekalni w celu uwzględnienia ich w najbliższym commicie. Używamy kropki '.' aby dodać wszystkie pliki.

> git rm --cached *file*|.

Usunięcie pliku lub plików z poczekalni . Pliki te nie zostaną umieszczone w commicie.

Nazwą ***poczekalnia*** określaliśmy w trakcie kursu miejsce oficjalnie nazywane ***staging area.***

Wycofywanie zmian

> git checkout *id*|*branch*

Powrót do dowlonego commita w trybie tylko do odczytu lub do wybranego brancha.

> git revert *id*

Usunięcie wszystkich zmian z jednego commita poprzez stworzenie nowego "*odwracającego*" commita.

> git restore *file*|.

Usunięcie wszystkich zmian znajdujących się w pliku lub plikach. Nowe pliką zostaną usunięte.

> git reset --(soft|hard) *id*

Permanentne usunięcie commitów do wybranego miejsca w historii. Użycie flagi *hard usuwa również wszystkie zmiany pozostałe w plikach.*

Rozgałęzienia (branche)

> git branch -(d|D) (-a) *name*

Stworzenie bądź usunięcie brancha.

> git checkout -b *name*

Stworzenie i automatyczna zmiana brancha.

Mergowanie (i konflikty)

> git merge *name*

Dołączanie zmian z innego brancha do brancha na którym aktualnie się znajdujemy.

Po rozwiązaniu konfliktów należy kontynuować merge za pomocą komendy **git commit** (bez flagi *-m*). Gdy otworzy nam się edytor **vim,** zamykamy go wpisując **:wq** i klikając Enter.

FRONTSCHOOL

GIT CHEAT SHEET

Praca z GitHub

Wysyłanie zmian

> git remote add *alias url*

Utworzenie aliasu dla zdalnego repozytorium. Powszechnie używanym aliasem jest *origin*.

> git push (*url*|*alias*) (*branch*)

Wysyłanie commitów do zdalnego repozytorium podając jego url albo alias oraz nazwę brancha.

Pobieranie zmian

> git clone *url*

Pobieranie całości zdalnego repozytorium na nasz komputer.

> git pull (*url*|*alias*)

Pobranie najnowszych zmian z danego brancha zdalnego repozytorium.

> git fetch (*url*|*alias*) (*branch*)

Pobieranie informacji o aktualnym statusie zdalnego repozytorium bez pobierania zmian.

> git remote -v

Podgląd aliasów utworzonych dla zdalnego repozytorium.

BONUS: Podstawowe komendy wiersza poleceń

> cd *folder*|..|\

Przenoszenie się pomiędzy folderami.
Przykłady użycia:

```
cd folder1/folder2/ - zagłębianie się w foldery
cd ../../ - przechodzenie do wyższych folderów
cd \ - przeniesienie się na dysk (np. C:\)
```

> *dysk:*

Zmiana dysku twardego.

```
d: - zmiana dysku na dysk D:\
```

> mkdir *name*

Stworzenie nowego katalogu.

> dir

Wyświetlenie zawartości katalogu.

> rmdir *name*

Usunięcie wybranego katalogu.

> code (.)

Uruchomienie VS Code. Dodanie kropki otworzy również aktualny folder wewnątrz VS Code.

> explorer (.)

Uruchomienie eksploratora plików. Dodanie kropki otworzy go w aktualnym folderze.

> cls

Wyczyszczenie okna wiersza poleceń